

SSSSSSSSSSSS	DDDDDDDDDDDD	AAAAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAAAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAAAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSSSSSSSSS	DDD	AAA
SSSSSSSSSS	DDD	AAA
SSSSSSSSSS	DDD	AAA
SSS	DDD	AAAAA
SSS	DDD	AAAAA
SSS	DDD	AAAAA
SSS	DDD	AAA
SSS	DDD	AAA
SSS	DDD	AAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAA
SSSSSSSSSSSS	DDDDDDDDDDDD	AAA

```
CCCCCCCC  RRRRRRRR  AAAAAA  SSSSSSSS  HH  HH
CCCCCCCC  RRRRRRRR  AAAAAA  SSSSSSSS  HH  HH
CC         RR      RR  AA      AA  SS      HH  HH
CC         RR      RR  AA      AA  SS      HH  HH
CC         RR      RR  AA      AA  SS      HH  HH
CC         RRRRRRRR  AA      AA  SSSSSS  HHHHHHHHHH
CC         RRRRRRRR  AA      AA  SSSSSS  HHHHHHHHHH
CC         RR  RR  AAAAAAAAAA  SS      HH  HH
CC         RR  RR  AAAAAAAAAA  SS      HH  HH
CC         RR  RR  AA      AA  SS      HH  HH
CC         RR  RR  AA      AA  SSSSSSSS  HH  HH
CCCCCCCC  RR      RR  AA      AA  SSSSSSSS  HH  HH
CCCCCCCC  RR      RR  AA      AA  SSSSSSSS  HH  HH
.....
```

```
LL         IIIIII  SSSSSSSS
LL         IIIIII  SSSSSSSS
LL         II      SS
LL         II      SS
LL         II      SS
LL         II      SS
LL         II      SSSSSS
LL         II      SSSSSS
LL         II      SS
LL         II      SS
LL         II      SS
LL         II      SS
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS
```


(1)	2	COPYRIGHT NOTICE
(1)	29	PROGRAM DESCRIPTION
(2)	63	DECLARATIONS
(3)	76	STORAGE DEFINITIONS
(4)	92	READ-ONLY DATA DEFINITIONS
(5)	113	DISPLAY CRASH -- DISPLAY CRASH INFORMATION
(6)	353	PRINT HEADER -- PRINT THE DUMP FILE HEADER BLOCKS
(7)	394	GET_DUMP_INFO -- GET DUMP HEADER INFO

```

0000 1      .TITLE  CRASH  DISPLAY CRASH RELATED INFORMATION
0000 2      .SBTTL  COPYRIGHT NOTICE
0000 3      .IDENT  'V04-000'
0000 4      :
0000 5      :*****
0000 6      :
0000 7      :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      :*  ALL RIGHTS RESERVED.
0000 10     :
0000 11     :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     :*  TRANSFERRED.
0000 17     :
0000 18     :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     :*  CORPORATION.
0000 21     :
0000 22     :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     :
0000 25     :
0000 26     :*****
0000 27     :

```



```

0000 29      .SBTTL  PROGRAM DESCRIPTION
0000 30      :++
0000 31      FACILITY
0000 32      :
0000 33      SYSTEM DUMP ANALYZER
0000 34      :
0000 35      ABSTRACT
0000 36      :
0000 37      THIS MODULE CONTAINS ROUTINES TO FORMAT INFORMATION
0000 38      SAVED AT CRASH TIME.
0000 39      :
0000 40      ENVIRONMENT
0000 41      :
0000 42      NATIVE MODE, USER MODE
0000 43      :
0000 44      AUTHOR
0000 45      :
0000 46      TIM HALVORSEN, JULY 1978
0000 47      :
0000 48      MODIFIED BY
0000 49      :
0000 50      V03-003 EMB0106      Ellen M. Batbouta      7-Jun-1984
0000 51      Increase the max size of a file name from 64 bytes to
0000 52      255 bytes.
0000 53      :
0000 54      V03-002 ROW0257      Ralph O. Weber      17-NOV-1983
0000 55      Cause SHOW CRASH to display the SCS node name, if one exists.
0000 56      :
0000 57      V03-001 TCM0001      Trudy C. Matthews      27-Jul-1983
0000 58      Added comment saying why the CPUDISP macro can't be used for
0000 59      some CPU-dependent code in this module.
0000 60      :
0000 61      :--

```

0000	63	.SBTTL	DECLARATIONS	
0000	64	:		
0000	65	:		
0000	66	:		
0000	67		\$DMPDEF	: DUMP FILE DEFINITIONS
0000	68		\$EMBDEF <CR,ET,HD>	: CRASHDUMP ERROR LOG ENTRY
0000	69		\$ERLDEF	: ERROR LOG DEFINITIONS
0000	70		\$PCBDEF	: PROCESS CONTROL BLOCK
0000	71		\$PHDDEF	: PROCESS HEADER DEFINITIONS
0000	72		\$IFDDEF	: IMAGE FILE DESCRIPTOR
0000	73		\$PSLDEF	: PROGRAM STATUS LONGWORD
0000	74		\$SBDEF	: SYSTEM BLOCK


```

0000 76 .SBTTL STORAGE DEFINITIONS
0000 77
0000 78 :
0000 79 : STORAGE DEFINITIONS
0000 80 :
0000 81
00000000 82 .PSECT SDADATA,NOEXE,WRT
0000 83
00000008 0000 84 ERLPTR::
0000 85 .BLKL 2 ; ADDRESS OF ERROR LOG ENTRY
0000 86
0000 87
00000000 88 .PSECT CRASH,EXE,NOWRT,LONG
0000 89
0000 90 .DEFAULT DISPLACEMENT, LONG

```

```

0000 92 .SBTTL READ-ONLY DATA DEFINITIONS
0000 93
0000 94 :
0000 95 :
0000 96 :
0000 97 :
0000 98 :
0000 99 :
0000 100 :
0000 101 :
0000 102 :
0000 103 RESTART_BUGS:
00000000: 0000 104 .LONG BUG$_IVLISTK ; INVALID INTERRUPT STACK
00000000: 0004 105 .LONG BUG$_DBLERR ; DOUBLE ERROR HALT
00000000: 0008 106 .LONG BUG$_HALT ; HALT INSTRUCTION
00000000: 000C 107 .LONG BUG$_ILLVEC ; ILLEGAL VECTOR CODE
00000000: 0010 108 .LONG BUG$_NOUSRWCS ; NO USER WCS FOR VECTOR
00000000: 0014 109 .LONG BUG$_ERRHALT ; ERROR PENDING ON HALT
00000000: 0018 110 .LONG BUG$_CHMONIS ; CHM ON INTERRUPT STACK
00000000: 001C 111 .LONG 0 ; --- END OF TABLE

```



```
0020 113 .SBTTL DISPLAY_CRASH -- DISPLAY CRASH INFORMATION
0020 114 :---
0020 115 :
0020 116 DISPLAY_CRASH
0020 117 :
0020 118 THIS ROUTINE DISPLAYS ALL RELATED INFORMATION REGARDING
0020 119 THE SAVED STATE OF THE PROCESSOR AT THE TIME OF THE
0020 120 SYSTEM BUGCHECK EXCEPTION.
0020 121 :
0020 122 INPUTS:
0020 123 :
0020 124 NONE
0020 125 :
0020 126 OUTPUTS:
0020 127 :
0020 128 NONE
0020 129 :
0020 130 :---
0020 131 :
0020 132 .ENABL LSB
0020 133 :
020C 0020 134 .ENTRY DISPLAY_CRASH,-
0022 135 ^M<R2,R3,R9>
0022 136 :
0022 137 SUBHD <System crash information> ; SET NEW HEADING
002F 138 SKIP PAGE
0036 139 :
59 00000000'EF D0 0036 140 MOVL ERLPTR,R9 ; ADDRESS OF ERROR LOG ENTRY
003D 141 ALLOC 24,R2 ; ALLOCATE SPACE FOR DATE/TIME
0047 142 $ASCTIM,S TIMADR=EMBSQ_CR_TIME(R9),TIMBUF=(R2)
52 DD 0057 143 PUSHL R2
0059 144 PRINT 1,<Time of system crash: !AS>
0066 145 :
0066 146 SKIP 2
006F 147 GETMEM @SYSSGQ_VERSION,-(SP) ; GET SYSTEM VERSION
5E DD 007F 148 PUSHL SP ; ADDRESS OF STRING
04 DD 0081 149 PUSHL #4 ; LENGTH OF STRING
0083 150 PRINT 3,<Version of system: VAX/VMS VERSION !AD>
53 00000000'GF D0 0090 151 MOVL G^SCSSGA_LOCALSB,R3 ; Get address of local system block
5E 10 C2 0097 152 SUBL #SBSS_NODENAME, SP ; Make scratch space for node name
52 5E D0 009A 153 MOVL SP,R2 ; Save address of scratch
009D 154 GETMEM SB$T_NODENAME(R3),- ; Get node name
009D 155 (R2),#SBSS_NODENAME
62 95 00AB 156 TSTB (R2) ; Is the node name null?
18 13 00AD 157 BEQL 15$ ; Branch if null node name
00AF 158 SKIP 2
52 DD 00B8 159 PUSHL R2 ; Push node name copy address
00BA 160 PRINT 1,<VAXcluster node name: !AC>
00C7 161 15$:
00C7 162 :
50 00F4 C9 FD 8F 78 00C7 163 ASHL #-3,EMBSL_CR_CODE(R9),R0 ; MESSAGE NUMBER
28 13 00CE 164 BEQL 25$ ; SKIP IF NO MESSAGE
51 00000000'EF 9E 00D0 165 MOVAB L^BUG$T_MESSAGES,R1 ; ADDRESS OF MESSAGES
00D7 166 20$:
52 81 9A 00D7 167 MOVZBL (R1)+,R2 ; LENGTH OF MESSAGE
51 52 C0 00DA 168 ADDL2 R2,R1 ; SKIP TO NEXT MESSAGE
F7 50 F5 00DD 169 SOBGTR R0,20$ ; LOOP UNTIL FOUND
```



```

51 DD 00E0 170 PUSHL R1 ; ADDRESS OF BUGCHECK MESSAGE
    DD 00E2 171 SKIP 2
    DD 00EB 172 PRINT 1,<Reason for BUGCHECK exception: !AC>
    DD 00F8 173
    DD 00F8 174 25$:
    DD 00F8 175 GETMEM @SCH$GL_CURPCB ; GET CURRENT PROCESS'S PCB
30 50 E9 0105 176 BLBC R0,26$ ; BRANCH IF DATA NOT AVAILABLE
    7E 7C 0108 177 CLRQ -(SP) ; INITIALIZE A BUFFER TO HOLD
    7E 7C 010A 178 CLRQ -(SP) ; THE CURRENT PROCESS'S NAME
52 5E D0 010C 179 MOVL SP,R2 ; REMEMBER ADDRESS OF BUFFER
    DD 010F 180 GETMEM PCB$T_LNAME(R1),(R2),#16 ; GET CURRENT PROCESS'S NAME
    DD 011D 181 PUSHL R2 ; PROCESS NAME
    DD 011F 182 SKIP 2
    DD 0128 183 PRINT 1,<Process currently executing: !AC>
    SE 10 C0 0135 184 ADDL #16,SP ; CLEAN BUFFER OFF STACK
    DD 0138 185 26$:
    DD 0138 186
00000000'EF 00 FB 0138 187 CALLS #0,CURPROC ; SET TO CURRENT PROCESS
51 00000000'EF D0 013F 188 MOVL MMG$IMGHDRBUF,R1 ; ADDRESS OF HEADER BUFFER
    DD 0146 189 TRYMEM 4(R1),R2 ; GET ADDRESS OF IFD
    DD 0153 190 BLBC R0,30$ ; IF NOT AVAILABLE
    DD 0156 191 TRYMEM IFD$W_FILNAMOFF(R1) ; GET OFFSET TO FILE NAME
    DD 0160 192 BLBC R0,30$ ; BRANCH IF NOT AVAILABLE
    51 51 32 0163 193 CVTWL R1,R1 ; CONVERT TO LONGWORD
5E 000000FF 8F C2 0166 194 SUBL #255,SP ; ALLOCATE BUFFER FOR FILESPEC
    50 5E D0 016D 195 MOVL SP,R0
    DD 0170 196 TRYMEM (R2)[R1],(R0),#255 ; GET ASCII IMAGE FILE NAME
    DD 0182 197 BLBC R0,30$ ; BRANCH IF NOT AVAILABLE
    DD 0185 198 PUSHL SP
    DD 0187 199 SKIP 2
    DD 0190 200 PRINT 1+<255/4>,<Current image file: !AC>
    DD 01A1 201
    DD 01A1 202 30$:
7E 64 A9 05 10 EF 01A1 203 EXTZV #PSL$V_IPL,#PSL$S_IPL,EMB$CR_PSL(R9),-(SP)
    DD 01A7 204 SKIP 2
    DD 01B0 205 PRINT 1,<Current IPL: !UL (decimal)>
    DD 01BD 206
    DD 01BD 207 SKIP 5
    DD 01C6 208 ENSURE 7
    DD 01DE 209 PRINT 0,<General registers:>
    DD 01EB 210 SKIP 1
    DD 01F4 211 :
    DD 01F4 212 :
    DD 01F4 213 :
    DD 01F4 214 :
    DD 01F4 215 :
50 00F4 C9 07 CB 01F4 216 BICL3 #7,EMB$CR_CODE(R9),R0 ; BUGCHECK CODE (CLEAR FLAG)
    51 FE02 CF DE 01FA 217 MOVAL RESTART_BUGS,R1 ; TABLE OF RESTART BUGCHECKS
    DD 01FF 218 40$:
    DD 01FF 219 CMPL R0,(R1)+ ; CHECK IF MATCHES
    DD 0202 220 BEQL 45$ ; BRANCH IF SO
    DD 0204 221 TSTL (R1) ; END OF TABLE?
    DD 0206 222 BNEQ 40$ ; CONTINUE UNTIL DONE
    DD 0208 223 BRB 50$ ; PRINT REGISTERS
    DD 020A 224 45$:
    DD 020A 225 PUSHL EMB$CR_R11(R9) ; PSL
    DD 020D 226 PUSHL EMB$CR_R10(R9) ; PC
```



```
0210 227 PRINT 2,<!_PC = !XL PSL = !XL>
021D 228 SKIP 1
0226 229 PRINT 0,<!_Remaining registers not available -- wiped out by console>
0233 230 SKIP 1
023C 231 PRINT 0,<Processor registers:>
0249 232 SKIP 1
0084 C9 DD 0252 233 PUSHL EMB$CR_SCBB(R9)
7C A9 DD 0256 234 PUSHL EMB$CR_SLR(R9)
78 A9 DD 0259 235 PUSHL EMB$CR_SBR(R9)
025C 236 PRINT 1,<!_SBR = !XL>
0269 237 PRINT 1,<!_SLR = !XL>
0276 238 PRINT 1,<!_SCBB = !XL>
021F 31 0283 239 BRW 60$ ; PRINT KSP-ISP REGISTERS
0286 240 50$:
30 A9 DD 0286 241 PUSHL EMB$CR_R3(R9)
2C A9 DD 0289 242 PUSHL EMB$CR_R2(R9)
28 A9 DD 028C 243 PUSHL EMB$CR_R1(R9)
24 A9 DD 028F 244 PUSHL EMB$CR_R0(R9)
0292 245 PRINT 4,<!_R0 = !XL R1 = !XL R2 = !XL R3 = !XL>
40 A9 DD 029F 246 PUSHL EMB$CR_R7(R9)
3C A9 DD 02A2 247 PUSHL EMB$CR_R6(R9)
38 A9 DD 02A5 248 PUSHL EMB$CR_R5(R9)
34 A9 DD 02A8 249 PUSHL EMB$CR_R4(R9)
02AB 250 PRINT 4,<!_R4 = !XL R5 = !XL R6 = !XL R7 = !XL>
50 A9 DD 02B8 251 PUSHL EMB$CR_R11(R9)
4C A9 DD 02BB 252 PUSHL EMB$CR_R10(R9)
48 A9 DD 02BE 253 PUSHL EMB$CR_R9(R9)
44 A9 DD 02C1 254 PUSHL EMB$CR_R8(R9)
02C4 255 PRINT 4,<!_R8 = !XL R9 = !XL R10 = !XL R11 = !XL>
60 A9 DD 02D1 256 PUSHL EMB$CR_PC(R9)
5C A9 DD 02D4 257 PUSHL EMB$CR_SP(R9)
58 A9 DD 02D7 258 PUSHL EMB$CR_FP(R9)
54 A9 DD 02DA 259 PUSHL EMB$CR_AP(R9)
02DD 260 PRINT 4,<!_AP = !XL FP = !XL SP = !XL PC = !XL>
64 A9 DD 02EA 261 PUSHL EMB$CR_PSL(R9)
02ED 262 PRINT 1,<!_PSL = !XL>
02FA 263 SKIP 4
0303 264 ENSURE 10
031B 265 PRINT 0,<Processor registers:>
0328 266 SKIP 1
0331 267 :
0331 268 : We can't use the CPUDISP macro here because we have to get the EXE$GB_CPUTYPE
0331 269 : value from the appropriate dump file.
0331 270 :
0331 271 GETMEM @EXE$GB_CPUTYPE ; GET TYPE OF CPU
09 50 E9 033E 272 BLBC R0,780$ ; IF NOT FOUND, ASSUME 11/780
0341 273 CASE R1,TYPE=B,- ; DISPATCH ON CPU TYPE
0341 274 LIMIT=#PR$SID_TYP780,- ;
0341 275 <780$,- ; 11/780
0341 276 750$> ; 11/750
034A 277 : ALL OTHERS USE 11/780
034A 278 :
034A 279 : 11/780 INTERNAL REGISTERS
034A 280 :
009C C9 DD 034A 281 780$: PUSHL EMB$CR_ACCS(R9)
0080 C9 DD 034E 282 PUSHL EMB$CR_PCBB(R9)
68 A9 DD 0352 283 PUSHL EMB$CR_POBR(R9)
```



```
00A0 C9 DD 0355 284 PRINT 3,<! POBR = !XL PCBB = !XL ACCS = !XL>
0084 C9 DD 0362 285 PUSHL EMB$[CR-SBIFS(R9)
6C A9 DD 0366 286 PUSHL EMB$[CR-SCBB(R9)
DD 036A 287 PUSHL EMB$[CR-POLR(R9)
00A4 C9 DD 036D 288 PRINT 3,<! POLR = !XL SCBB = !XL SBIFS = !XL>
0088 C9 DD 037A 289 PUSHL EMB$[CR-SBISC(R9)
70 A9 DD 037E 290 PUSHL EMB$[CR-ASTLVL(R9)
DD 0382 291 PUSHL EMB$[CR-P1BR(R9)
00A8 C9 DD 0385 292 PRINT 3,<! P1BR = !XL ASTLVL = !XL SBISC = !XL>
008C C9 DD 0392 293 PUSHL EMB$[CR-SBIMT(R9)
74 A9 DD 0396 294 PUSHL EMB$[CR-SISR(R9)
DD 039A 295 PUSHL EMB$[CR-P1LR(R9)
00AC C9 DD 039D 296 PRINT 3,<! P1LR = !XL SISR = !XL SBIMT = !XL>
0090 C9 DD 03AA 297 PUSHL EMB$[CR-SBIER(R9)
78 A9 DD 03AE 298 PUSHL EMB$[CR-ICCS(R9)
DD 03B2 299 PUSHL EMB$[CR-SBR(R9)
00B0 C9 DD 03B5 300 PRINT 3,<! SBR = !XL ICCS = !XL SBIER = !XL>
0094 C9 DD 03C2 301 PUSHL EMB$[CR-SBITA(R9)
7C A9 DD 03C6 302 PUSHL EMB$[CR-ICR(R9)
DD 03CA 303 PUSHL EMB$[CR-SLR(R9)
00B4 C9 DD 03CD 304 PRINT 3,<! SLR = !XL ICR = !XL SBITA = !XL>
0098 C9 DD 03DA 305 PUSHL EMB$[CR-SBIS(R9)
DD 03DE 306 PUSHL EMB$[CR-TODR(R9)
DD 03E2 307 PRINT 2,<! - TODR = !XL SBIS = !XL>
00AA 31 03EF 308 SKIP 1
03F8 309 BRW 60$
03FB 310
03FB 311 11/750 INTERNAL REGISTERS
03FB 312 750$:
009C C9 DD 03FB 313 PUSHL EMB$[CR-ACCS(R9)
0080 C9 DD 03FF 314 PUSHL EMB$[CR-PCBB(R9)
68 A9 DD 0403 315 PUSHL EMB$[CR-POBR(R9)
DD 0406 316 PRINT 3,<! POBR = !XL PCBB = !XL ACCS = !XL>
00A0 C9 DD 0413 317 PUSHL EMB$[CR-TBDR(R9)
0084 C9 DD 0417 318 PUSHL EMB$[CR-SCBB(R9)
6C A9 DD 041B 319 PUSHL EMB$[CR-POLR(R9)
DD 041E 320 PRINT 3,<! POLR = !XL SCBB = !XL TBDR = !XL>
00A4 C9 DD 042B 321 PUSHL EMB$[CR-CADR(R9)
0088 C9 DD 042F 322 PUSHL EMB$[CR-ASTLVL(R9)
70 A9 DD 0433 323 PUSHL EMB$[CR-P1BR(R9)
DD 0436 324 PRINT 3,<! P1BR = !XL ASTLVL = !XL CADR = !XL>
00A8 C9 DD 0443 325 PUSHL EMB$[CR-MCESR(R9)
008C C9 DD 0447 326 PUSHL EMB$[CR-SISR(R9)
74 A9 DD 044B 327 PUSHL EMB$[CR-P1LR(R9)
DD 044E 328 PRINT 3,<! P1LR = !XL SISR = !XL MCESR = !XL>
00AC C9 DD 045B 329 PUSHL EMB$[CR-CAER(R9)
0090 C9 DD 045F 330 PUSHL EMB$[CR-ICCS(R9)
78 A9 DD 0463 331 PUSHL EMB$[CR-SBR(R9)
DD 0466 332 PRINT 3,<! SBR = !XL ICCS = !XL CAER = !XL>
00B0 C9 DD 0473 333 PUSHL EMB$[CR-CMIERR(R9)
0094 C9 DD 0477 334 PUSHL EMB$[CR-ICR(R9)
7C A9 DD 047B 335 PUSHL EMB$[CR-SLR(R9)
DD 047E 336 PRINT 3,<! SLR = !XL ICR = !XL CMIERR = !XL>
0098 C9 DD 048B 337 PUSHL EMB$[CR-TODR(R9)
DD 048F 338 PRINT 1,<! - TODR = !XL>
DD 049C 339 SKIP 1
04A5 340 60$:
```


CRASH
V04-000

DISPLAY CRASH RELATED INFORMATION F 7
DISPLAY_CRASH -- DISPLAY CRASH INFORMATI 16-SEP-1984 01:25:55 VAX/VMS Macro V04-00
5-SEP-1984 03:32:04 [SDA.SRC]CRASH.MAR;1

Page 10
(5)

20	A9	DD	04A5	341	PUSHL	EMB\$CR_ISP(R9)
			04A8	342	PRINT	1,<!--ISP-->=!XL>
1C	A9	DD	04B5	343	PUSHL	EMB\$CR_USP(R9)
18	A9	DD	04B8	344	PUSHL	EMB\$CR_SSP(R9)
14	A9	DD	04BB	345	PUSHL	EMB\$CR_ESP(R9)
10	A9	DD	04BE	346	PUSHL	EMB\$CR_KSP(R9)
			04C1	347	PRINT	1,<!--RSP-->=!XL>
			04CE	348	PRINT	1,<!--ESP-->=!XL>
			04DB	349	PRINT	1,<!--SSP-->=!XL>
			04E8	350	PRINT	1,<!--USP-->=!XL>
04			04F5	351	RET	

LIE
V04

```
04F6 353 .SBTTL PRINT_HEADER -- PRINT THE DUMP FILE HEADER BLOCKS
04F6 354 :---
04F6 355 :
04F6 356 PRINT_HEADER
04F6 357 :
04F6 358 DUMP THE CONTENTS OF THE DUMP FILE HEADER BLOCKS
04F6 359 :
04F6 360 INPUTS:
04F6 361 :
04F6 362 DUMP_HEADER = DUMP FILE HEADER (3 BLOCKS)
04F6 363 :
04F6 364 OUTPUTS:
04F6 365 :
04F6 366 NONE
04F6 367 :
04F6 368 :---
04F6 369 :
04F6 370 .ENABL LSB
04F6 371 :
04F6 372 PRINT_HEADER::
0004 04F6 373 .WORD ^M<R2>
04F6 374 :
04F6 375 SUBHD <Dump file header>
0505 376 SKIP PAGE
52 00000000'EF DE 050C 377 MOVAL DUMP_HEADER,R2 ; STARTING ADDRESS
53 00000600 8F D0 0513 378 MOVL #3*5T2,R3 ; LENGTH TO DUMP
051A 379 10$:
52 DD 051A 380 PUSHL R2 ; DUMP ADDRESS
52 DD 051C 381 PUSHL R2 ; ADDRESS OF ASCII STRING
20 DD 051E 382 PUSHL #32 ; LENGTH OF STRING
50 08 D0 0520 383 MOVL #8,R0 ; REPEAT COUNT
0523 384 20$:
82 DD 0523 385 PUSHL (R2)+ ; PUSH NEXT 8 LONGWORDS
FB 50 F5 0525 386 SOBGTR R0,20$ ; PUSH NEXT 8 LONGWORDS
0528 387 PRINT 11,<!XL !XL !XL !XL !XL !XL !XL !XL !XL !XL>
53 20 C2 0535 388 SUBL2 #32,R3 ; DECREMENT LENGTH
E0 14 0538 389 BGTR 10$ ; CONTINUE UNTIL DONE
04 053A 390 RET
053B 391 :
053B 392 .DSABL LSB
```



```
053B 394 .SBTTL GET_DUMP_INFO -- GET DUMP HEADER INFO
053B 395 ---
053B 396
053B 397 GET_DUMP_INFO
053B 398
053B 399 FIND THE ERROR LOG ENTRY AND SAVE IT. IF THE
053B 400 ENTRY CANNOT BE FOUND, GENERATE A DUMMY ENTRY.
053B 401
053B 402 INPUTS:
053B 403
053B 404 NONE
053B 405
053B 406 OUTPUTS:
053B 407
053B 408 ERLPTR = ADDRESS OF ERROR LOG ENTRY
053B 409
053B 410 ---
053B 411 .ENABL LSB
053B 412
0304 053B 413 GET_DUMP_INFO::
053B 414 .WORD ^M<R2,R8,R9>
053D 415
053D 416
053D 417 SET FLAGS DESCRIBING THE VERSION OF THE SYSTEM
053D 418
053D 419 CLRL VERSION_FLAGS ; PRESET FLAGS LONGWORD
0543 420 GETMEM @SYSSGQ-VERSION ; READ THE SYSTEM VERSION
0550 421 CMPW R1,#^A^T. ; 1.0 EXECUTIVE? (1.01,1.5 ALSO)
0555 422 BEQL 2$ ; BRANCH IF NOT
0557 423 MOVL #1,VERSION_FLAGS ; INDICATE VERSION 2
055E 424 2$:
055E 425 BLBC CURRENT_SYSTEM,3$ ; BRANCH IF EXAMINING DUMP FILE
0565 426 BRW 30$ ; SETUP FOR CURRENT SYSTEM
0568 427 3$:
0568 428 MOVAB DUMP_HEADER+DMP$SL CRASHERL,ERLPTR ; SET PTR TO ERROR LOG ENTRY
0573 429 CMPW DUMP_HEADER+DMP$W-DUMPVER,#2 ; IS ERR MSG HDR AT NEG OFFSETS?
057A 430 BLSS 1$ ; BR IF NO, (VMS V2 FORMAT)
057C 431 ADDL #EMB$K_LENGTH,ERLPTR ; POINT PAST THE ERRMSG HDR (VMS V3)
0583 432 1$:
0583 433 MOVL ERLPTR,R9 ; ADDRESS OF ERROR LOG ENTRY
058A 434
058A 435 THE FOLLOWING CODE ADJUSTS THE SAVED SP SINCE IT WAS
058A 436 NOT SAVED UNTIL AFTER THE PC,PSL WAS ADDED TO THE STACK.
058A 437
058A 438 4$: ADDL2 #2*4,EMB$CR_SP(R9) ; ADJUST DUE TO BUGCHECK
058E 439
058E 440 STORE THE CURRENT SP (R14) INTO THE CORRESPONDING PROCESSOR
058E 441 REGISTER CORRESPONDING TO THE ACCESS MODE ONLY IF KERNEL OR
058E 442 INTERRUPT STACK. IF OTHER STACK, THEN USE THE PROCESSOR
058E 443 REGISTER BECAUSE BUGCHECK HAS TO MOVE THE INFO TO THE KERNEL
058E 444 MODE STACK AND IN THE PROCESS, WIPES THE SP REGISTER.
058E 445
058E 446
51 64 A9 02 18 EF 058E 446 EXTZV #PSL$V_CURMOD,#PSL$S_CURMOD,EMB$CR_PSL(R9),R1
0594 447 BEQL 10$ ; BRANCH IF KERNEL/INTERRUPT
5C A9 10 A9 41 D0 0596 448 MOVL EMB$CR_KSP(R9)[R1],EMB$CR_SP(R9) ; FIX SP VALUE
059C 449 BRB 6$
08 64 A9 1A E0 059E 450 10$: BBS #PSL$V_IS,EMB$CR_PSL(R9),5$ ; BRANCH IF ISP
```

```
10 A941 5C A9 D0 05A3 451      MOVL  EMB$$_CR_SP(R9),EMB$$_CR_KSP(R9)[R1] ; R14 TO PR[MODE+1]
      05 11 05A9 452      BRB    6$
20 A9 5C A9 D0 05AB 453 5$:    MOVL  EMB$$_CR_SP(R9),EMB$$_CR_ISP(R9) ; R14 TO PR[0]
      05 05B0 454 6$:
      05 05B0 455 :
      05 05B0 456 :
      05 05B0 457 :
      05 05B0 458 :
      05 05B0 459 :
      05 05B0 460 :
43 50 E9 05BD 461      GETMEM @SCH$$_GL_CURPCB ; ADDRESS OF CURRENT PCB
      05 05C0 462      BLBC    R0,21$ ; BRANCH IF CANNOT READ
33 50 E9 05CD 463      GETMEM PCB$$_L_PHD(R1),R2 ; GET ADDRESS OF CURRENT PHD
      52 D5 05D0 464      BLBC    R0,21$ ; BRANCH IF CANNOT READ
      2F 13 05D2 465      TSTL    R2 ; PHD VALID?
      05D4 466      BEQL    21$ ; SKIP IF NOT
      05E3 467      PUTMEM  PHD$$_L_KSP(R2),EMB$$_CR_KSP(R9),#4*4 ; SET KSP - USP
      05F3 468      PUTMEM  PHD$$_L_R0(R2),EMB$$_CR_R0(R9),#14*4 ; SET R0 - R13
      04 0603 469 21$:    PUTMEM  PHD$$_L_PC(R2),EMB$$_CR_PC(R9),#2*4 ; SET PC,PSL
      0604 470      RET
      0604 471 :
      0604 472 :
      0604 473 :
      0604 474 :
58 00000000'EF 9E 0604 475 30$: MOVAB  DUMP_HEADER,R8 ; ADDRESS OF DUMP HEADER
      00 04 A8 00 E2 060B 476      BBSS  #DMP$$_V_OLDDUMP,DMP$$_L_FLAGS(R8),32$ ; FLAG DUMP ANALYZED
      0000010C 8F DD 0610 477 32$: PUSHL  #EMB$$_C_CR_LENGTH ; LENGTH OF CRASH LOG ENTRY
      00000000'EF 01 FB 0616 478      CALLS #1,ALLOCATE ; ALLOCATE STORAGE
      00000000'EF 51 D0 061D 479      MOVL  R1,ERLPTR ; SAVE ADDRESS OF ERL BUFFER
      59 51 D0 0624 480      MOVL  R1,R9 ; REFERENCE OFF R9
      0627 481      GETMEM  @EXE$$_GQ_SYSTIME,EMB$$_CR_TIME(R9),#8 ; SET DATE/TIME
      0639 482      GETMEM  @MMG$$_GL_SPTLEN,R2 ; LENGTH OF SPT IN LONGWORDS
      0649 483      GETMEM  @MMG$$_GL_SBR ; PHYSICAL ADDRESS OF SPT
      0656 484      MOVAL  (R1)[R2],R1 ; COMPUTE PHYSICAL MEMORY SIZE
      065A 485      ASHL  #-9,R1,PHYS_PAGES ; SAVE MEMORY SIZE IN PAGES
      0663 486      GETMEM  @SCH$$_GL_CURPCB,R2 ; GET ADDRESS OF CURRENT PCB
      0673 487      BLBC    R0,35$ ; BRANCH IF ERROR
      0676 488      GETMEM  PCB$$_L_PID(R2),EMB$$_CR_PID(R9),#4 ; GET PID OF CURRENT PROCESS
      0685 489      GETMEM  PCB$$_L_LNAME(R2),EMB$$_CR_LNAME(R9),#16 ; AND ALSO NAME
      0695 490 35$:      STATUS SUCCESS
      04 069C 491      RET
      069D 492
      069D 493      .DSABL  LSB
      069D 494
      069D 495      .END
```


CRASH
Symbol table

DISPLAY CRASH RELATED INFORMATION

J 7

16-SEP-1984 01:25:55 VAX/VMS Macro V04-00
5-SEP-1984 03:32:04 [SDA.SRC]CRASH.MAR;1

Page 14
(7)

ALLOCATE	*****	X	03
ARGS	= 00000003		
BUG\$T_MESSAGES	*****	X	03
BUG\$T_MONIS	*****	X	03
BUG\$T_DBLERR	*****	X	03
BUG\$T_ERRHALT	*****	X	03
BUG\$T_HALT	*****	X	03
BUG\$T_ILLVEC	*****	X	03
BUG\$T_IVLISTK	*****	X	03
BUG\$T_NOUSRWCS	*****	X	03
CURPROC	*****	X	03
CURRENT_SYSTEM	*****	X	03
DISPLAY_CRASH	00000020	RG	03
DMP\$T_CRASHERL	= 0000006C		
DMP\$T_FLAGS	= 00000004		
DMP\$V_OLDDUMP	= 00000000		
DMP\$W_DUMPVER	= 00000006		
DUMP_HEADER	*****	X	03
EMBSL_CR_LENGTH	= 0000010C		
EMBSL_LENGTH	= 00000004		
EMBSL_CR_ACCS	= 0000009C		
EMBSL_CR_AP	= 00000054		
EMBSL_CR_ASTLVL	= 00000088		
EMBSL_CR_CADR	= 000000A4		
EMBSL_CR_CAER	= 000000AC		
EMBSL_CR_CMIERR	= 000000B0		
EMBSL_CR_CODE	= 000000F4		
EMBSL_CR_ESP	= 00000014		
EMBSL_CR_FP	= 00000058		
EMBSL_CR_ICCS	= 00000090		
EMBSL_CR_ICR	= 00000094		
EMBSL_CR_ISP	= 00000020		
EMBSL_CR_KSP	= 00000010		
EMBSL_CR_MCESR	= 000000A8		
EMBSL_CR_POBR	= 00000068		
EMBSL_CR_POLR	= 0000006C		
EMBSL_CR_P1BR	= 00000070		
EMBSL_CR_P1LR	= 00000074		
EMBSL_CR_PC	= 00000060		
EMBSL_CR_PCBB	= 00000080		
EMBSL_CR_PID	= 000000F8		
EMBSL_CR_PSL	= 00000064		
EMBSL_CR_R0	= 00000024		
EMBSL_CR_R1	= 00000028		
EMBSL_CR_R10	= 0000004C		
EMBSL_CR_R11	= 00000050		
EMBSL_CR_R2	= 0000002C		
EMBSL_CR_R3	= 00000030		
EMBSL_CR_R4	= 00000034		
EMBSL_CR_R5	= 00000038		
EMBSL_CR_R6	= 0000003C		
EMBSL_CR_R7	= 00000040		
EMBSL_CR_R8	= 00000044		
EMBSL_CR_R9	= 00000048		
EMBSL_CR_SBIER	= 000000AC		
EMBSL_CR_SBIFS	= 000000A0		
EMBSL_CR_SBIMT	= 000000A8		

EMBSL_CR_SBIS	= 000000B4		
EMBSL_CR_SBISC	= 000000A4		
EMBSL_CR_SBITA	= 000000B0		
EMBSL_CR_SBR	= 00000078		
EMBSL_CR_SCBB	= 00000084		
EMBSL_CR_SISR	= 0000008C		
EMBSL_CR_SLR	= 0000007C		
EMBSL_CR_SP	= 0000005C		
EMBSL_CR_SSP	= 00000018		
EMBSL_CR_TBDR	= 000000A0		
EMBSL_CR_TODR	= 00000098		
EMBSL_CR_USP	= 0000001C		
EMBSQ_CR_TIME	= 00000006		
EMBST_CR_LNAME	= 000000FC		
ERLPTR	00000000	RG	02
EXESGB_CPUYPE	*****	X	03
EXESGQ_SYSTIME	*****	X	03
GETMEM	*****	X	03
GET_DUMP_INFO	0000053B	RG	03
IFDSW_FILNAMOFF	= 00000002		
LINE_COUNT	*****	X	03
MMG\$GL_SBR	*****	X	03
MMG\$GL_SPTLEN	*****	X	03
MMG\$IMGHDRBUF	*****	X	03
MSG\$ SUCCESS	*****	X	03
NEW PAGE	*****	X	03
PAGE_SIZE	*****	X	03
PCBSL_PHD	= 0000006C		
PCBSL_PID	= 00000060		
PCBST_LNAME	= 00000070		
PHDSL_KSP	= 00000078		
PHDSL_PC	= 000000C0		
PHDSL_RO	= 00000088		
PHYS_PAGES	*****	X	03
PRS_SID_TYP780	*****	X	03
PRINT	*****	X	03
PRINT_HEADER	000004F6	RG	03
PSL\$S_CURMOD	= 00000002		
PSL\$S_IPL	= 00000005		
PSL\$V_CURMOD	= 00000018		
PSL\$V_IPL	= 00000010		
PSL\$V_IS	= 0000001A		
PUTMEM	*****	X	03
RESTART BUGS	00000000	R	03
SB\$S_NODENAME	= 00000010		
SB\$T_NODENAME	= 00000044		
SCH\$GL_CURPCB	*****	X	03
SCS\$GA_LOCALSB	*****	X	03
SET HEADING	*****	X	03
SKIP LINES	*****	X	03
SYSSASCTIM	*****	GX	03
SYSSGQ_VERSION	*****	X	03
TRYMEM	*****	X	03
VERSION_FLAGS	*****	X	03

LIE
V04

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SDADATA	00000008 (8.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE
CRASH	0000069D (1693.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG
LITERALS	000006CC (1740.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.06	00:00:00.93
Command processing	146	00:00:00.48	00:00:03.66
Pass 1	297	00:00:05.83	00:00:24.15
Symbol table sort	0	00:00:00.61	00:00:01.00
Pass 2	111	00:00:01.50	00:00:05.27
Symbol table output	15	00:00:00.06	00:00:00.08
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	608	00:00:08.56	00:00:35.11

The working set limit was 1650 pages.
52887 bytes (104 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 573 non-local and 70 local symbols.
495 source lines were read in Pass 1, producing 32 object records in Pass 2.
31 pages of virtual memory were used to define 30 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
-\$255\$DUA28:[SDA.OBJ]SDALIB.MLB;1	10
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	10
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	27

745 GETS were required to define 27 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:CRASH/OBJ=OBJ\$:CRASH MSRC\$:CRASH/UPDATE=(ENH\$:CRASH)+EXECML\$/LIB+LIB\$:SDALIB/LIB

0351 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700
701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000